BlockchainLabs.NZ
Blockchain Solution Design & Development

Fox Token

audit report

Table of contents

Background

This audit report was undertaken by **BlockchainLabs.nz** for the purpose of providing feedback to **ShapeShift**.

Solidity contracts were provided by the ShapeShift development team at this commit [bd2f1f677e5956846aeefdab28d725291443366f] (private repo)

Open Zeppelin Framework Usage

The ShapeShift team has utilised the Open Zeppelin Token framework which can be found here. This framework is an open source framework which has been contributed to and audited by many teams from all over the world including us (here & here). This framework is in our estimation of the highest quality and free of any issues and defects.

We will verify that the ShapeShift team is using this framework completely and correctly. We will verify that any custom code is free of defects. We will rely on our previous audits and audits we have reviewed from other team for any code that is apart of the Open Zeppelin Token framework.

Scope

The following sources were in scope for static and dynamic analysis:

FOX.sol

Document structure

The report will include the following sections:

- Static analysis
- Dynamic analysis
- Observations
- Conclusion

Focus areas

Correctness	No correctness defects uncovered during static analysis? No implemented contract violations uncovered during execution? No other generic incorrect behaviour detected during execution? Adherence to adopted standards such as ERC20?
Testability	Test coverage across all functions and events? Test cases for both expected behaviour and failure modes? Settings for easy testing of a range of parameters? No reliance on nested callback functions or console logs? Avoidance of test scenarios calling other test scenarios?
Security	No presence of known security weaknesses? No funds at risk of malicious attempts to withdraw/transfer? No funds at risk of control fraud? Prevention of Integer Overflow or Underflow?
Best Practice	Explicit labeling for the visibility of functions and state variables? Proper management of gas limits and nested execution? The latest version of the Solidity compiler?

Issues

Severity description

Minor A defect that does not have a material impact on the contract execution and is likely to be subjective.

Moderate A defect that could impact the desired outcome of the contract execution in a specific scenario.

Major A defect that impacts the desired outcome of the contract execution or introduces a weakness that may be exploited.

Critical A defect that presents a significant security vulnerability or failure of the contract across a range of scenarios.

Minor

- None found

Moderate

- None found

Major

- None found

Critical

- None found

Conclusion

We have verified that the Shapeshift team is using the most up to date version of the popular and well audited ERC20 framework “OpenZeppelin” to base their contract on. We are satisfied that there are no security issues in this contract and are satisfied that ETH or ERC20 tokens are not at risk of being hacked or stolen through interaction with this contract.

Disclaimer

Our team uses our current understanding of the best practises for Solidity and Smart Contracts. Development in Solidity and for Blockchain is an emergering area of software engineering which still has a lot of room to grow, hence our current understanding of best practise may not find all of the issues in this code and design.

We have not analysed any of the assembly code generated by the Solidity compiler. We have not verified the deployment process and configurations of the contracts. We have only analysed the code outlined in the scope. We have not verified any of the claims made by any of the organisations behind this code.

Security audits do not warrant bug-free code. We encourage all users interacting with smart contract code to continue to analyse and inform themselves of any risks before interacting with any smart contracts.